Research and Innovation

Master's degree in Modelling for Science and Engineering

Regression trees and random forests for the Boston dataset.

Abstract:

In this report we have done a full analysis for the prediction of the average house price (MEDV) in the area of Boston Mass. We have obtained a model with a MSE of 45.055 for the regression tree model whereas for the random forest the best model has had a MSE of 22.739.

Author:

Joseba Hernández Bravo

Teacher:

Isabel Serra Mochales

Barcelona, January 1, 2022

Contents

Chapter 1

Introduction

In this report we will make a complete analysis of the Boston dataset. The dataset contains housing prices in the city of Boston , as well as socio-economic information on the neighbourhood wich they are located.

The dataset, downloaded from Kaggle [1], is composed of the following columns:

- CRIM - per capita crime rate by town
- ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS - proportion of non-retail business acres per town.
- CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- NOX - nitric oxides concentration (parts per 10 million)
- RM - average number of rooms per dwelling
- AGE - proportion of owner-occupied units built prior to 1940
- DIS - weighted distances to five Boston employment centres
- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per 10,000 dollars
- PTRATIO - pupil-teacher ratio by town
- B - 1000\$$(Bk - 0.63)^2$\$ where Bk is the proportion of blacks by town
- MEDV - Median value of owner-occupied homes in \$1000's

The questions we ask ourselves are the following: will we be able to fit a regression model that predicts house prices for future data? which variables have the greatest weight for the prediction of MEDV? which model best fits our needs?

Chapter 2

Strategy

In this repor we will make a complete analysis of the Boston dataset. To do so, we will follow the following steps:

- **Data acquisition** : we will download the data from a reliable source.
- **Preprossessing data**: once we download the data, we have to investigate the most relevant characteristics of the data we are going to work with. In this way we will be able to further eliminate or modify data that we believe may be detrimental to our objective.
- **Variable selection**: in this section we will select the most representative variables for our models using different statistical criteria.
- **Model building** In this section we will build our models. Based on the results we obtain, we will decide whether to optimise them using techniques such as hyper-parameter tuning. After that, we will validate our model using data reserved for it and we will compare the results obtained for each regression model: **decision trees** and **random forests**.
- **Results**: In this last section we will summarise the results obtained and give some ideas on how to improve them.

We have created a Git-hub repository for storing all the code used in this report.

The necessary libraries are contained in the *requirements.txt* text file and the work is explained step by step in the file *analysis.ipynb*.

Chapter 3

Data analysis

In this first section we will analyse the dataset and try to handle it in such a way that for the following sections we can get the maximum performance out of our models.

In total, we have 12 columns that we will use as features and each of them contains 506 different values. The first thing we will do is to observe how the values are distributed in these features.

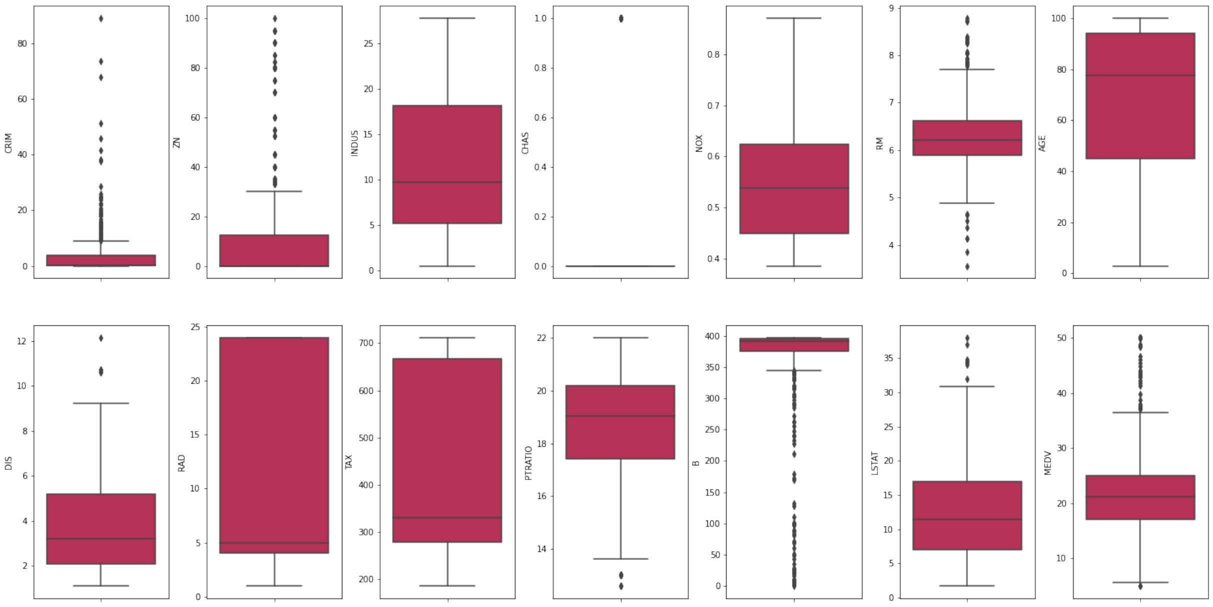


Figure 3.1: Box-plots of all the variables.

As we can see in 4.1, some of our features such a as CRIM, ZN, CHAS, RM, PTRATIO, B, LSTAT has outliers. However, the percentage of these is not the same in all cases. Indeed, CRIM, ZN and B variables show a high percentage of outliers compared to the rest. Since this data can be detrimental to our prediction, it is convenient to treat it in someway. One of the options is to discard all lines containing outliers. However, this procedure would cause us to lose a large amount of data. Another alternative is to replace the outliers by a statistic such as the mean, median... Although in this case the amount of data lost would be zero, we could end up with a model that is too biased.

Therefore, we have decided to use an intermediate strategy. That is, we have removed all data exceeding a set threshold (the lowest/highest half of the total lower/upper outiliers) and replaced all remaining outliers with the mean of each variable.

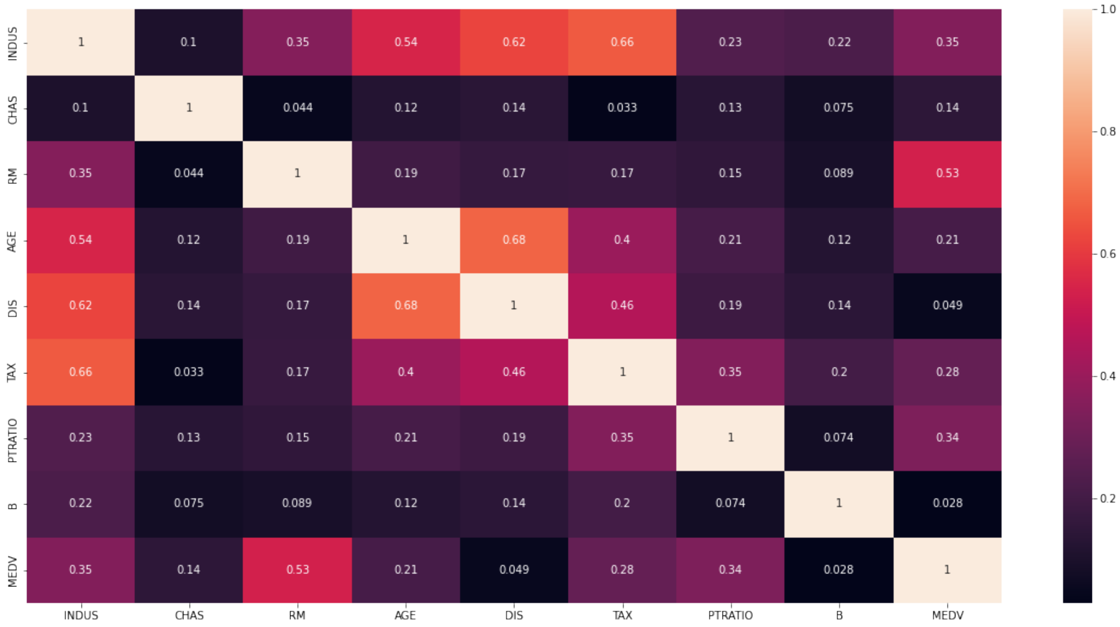
3.0.1 correlation

Another fundamental step when modifying our data for the proper functioning of the model is the choice of the characteristics that we are going to use.

The first step we have followed is to eliminate characteristics based on their p-value. That is, we have calculated the p-value for each characteristic and we have eliminated the cases in which the result has exceeded the threshold of 0.05. In this way, we went from having 12 variables to having 10.

After this, we have calculated the correlation matrix and we have decided to eliminate one of each pair of characteristics with a correlation higher or equal to 70%. The decision to eliminate variables was based on the correlation with the response variable. Thas it, we have compared in each case the correlation between each of the variables and MEDV.

Finally, the number of variables selected is 8 and the correlation matrix is the following:



	INDUS	CHAS	RM	AGE	DIS	TAX	PTRATIO	B	MEDV
INDUS	1	0.1	0.35	0.54	0.62	0.66	0.23	0.22	0.35
CHAS	0.1	1	0.044	0.12	0.14	0.033	0.13	0.075	0.14
RM	0.35	0.044	1	0.19	0.17	0.17	0.15	0.089	0.53
AGE	0.54	0.12	0.19	1	0.68	0.4	0.21	0.12	0.21
DIS	0.62	0.14	0.17	0.68	1	0.46	0.19	0.14	0.049
TAX	0.66	0.033	0.17	0.4	0.46	1	0.35	0.2	0.28
PTRATIO	0.23	0.13	0.15	0.21	0.19	0.35	1	0.074	0.34
B	0.22	0.075	0.089	0.12	0.14	0.2	0.074	1	0.028
MEDV	0.35	0.14	0.53	0.21	0.049	0.28	0.34	0.028	1

Figure 3.2: Last correlation matrix with the 8 selected variables.

Chapter 4

Model fitting

In this second section, we will explain the steps we have followed for the training and validation using the following models[1]:

- Decision trees.
- Regression random forest

4.0.1 Regression Trees

Decision trees are models that learn a hierarchy of if else questions [2]. Therefore, depending on the number of if/else (depth of our tree) that our model contains, we will be able to adjust better or worse to the expected result. We can say that, the precision of the model increases at the same time as the complexity. Thus, in an extreme case, we can imagine a tree with an answer for each question.

The main problem in this type of model is the *overfittin.* Therefore, in order to prevent this, we decided to follow a strategy called *pre-pruning.* This strategy is based on limiting the number of trees.

Once we have divided our dataset, in our case 30% for the training part and 70% , we proceed to test different combinations of parameters (such as the maximum depth of the tree) until we get the best result. In addition, since we have to take into account *overfitting*, for each combination of parameters we use cross-validation to reduce the contingency.

The metric we have used to evaluate our model is the mean squared error (MSE):

$$MSE = \frac{1}{N}\sum_{t=1}^{n}(y_i - \widehat{y_i})^2 \tag{4.1}$$

[1] the library used for both training and validation of the models has been **Sklearn**

We have conclude than in our case, the optimum number depth is 4. Here we can see how is the model complexity performing depending on the number of depth:

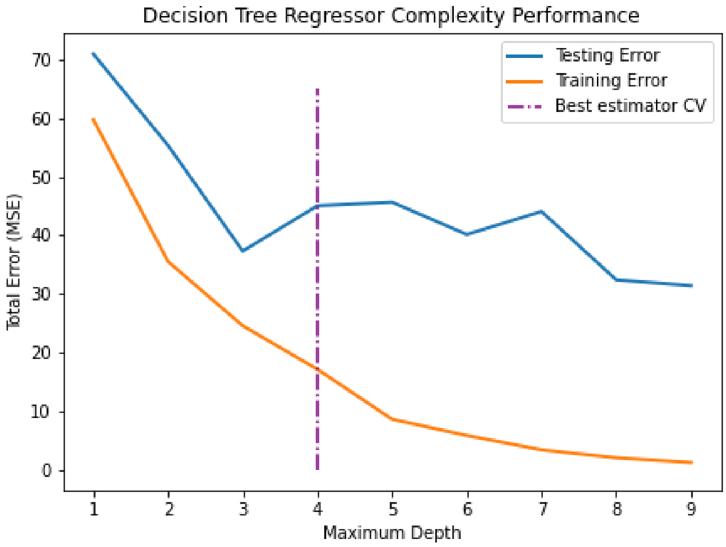


Figure 4.1: Total error MSE vs number of depth

And the performance of the model is:

MSE-s in and out of sample for a dummy model

Error	in-sample	out-sample
MSE	17.119	45.055

4.0.2 Random forest

In this second case, we have followed the same strategy. We have divided our dataset in 70% for training and 30% for test and we have trained our model using 10 trees and all the remaining parameters with their default values. Thus, we have achieved the following results:

MSE-s in and out of sample for a dummy model

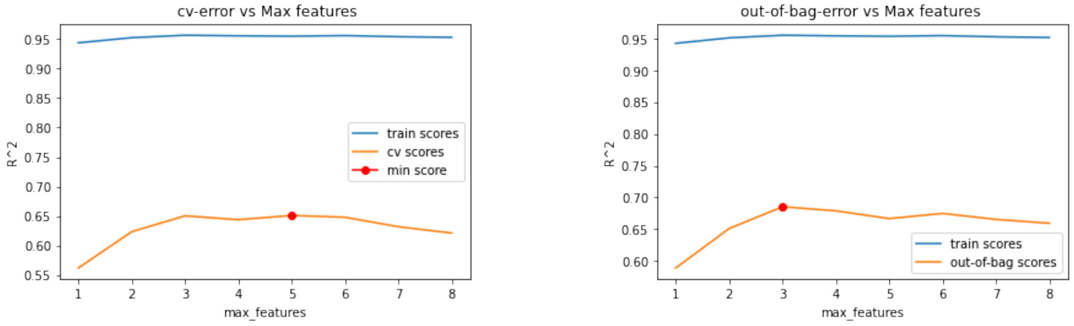
Error	in-sample	out-sample
MSE	6.029	32.775

We can see that, despite having used all the default parameters, the result we obtain in this case is better than that obtained with the previous model. However, we will make changes in the hyperparameters to see if we can optimise our model.

To find the optimal values of the hyperparameters, we will use two different metrics. On the one hand we will do the validation using Out-of-Bag[5] error and on the other hand, as in the case of the decision trees we will use the CV.

On the one hand, we have calculated the error in the model using different numbers of trees. In this way we have come to the conclusion that from a value between [71, 146] the error stabilizes. On the other hand, we have followed the same strategy leaving the number of trees constant at a value of 150 and varying the number of columns of train. In this way we obtain that the optimal number of variables to use is between [3,5].

Finally, once again, we have set the value of the number of trees to 150 and we have tried to make all possible combinations using a number of iterations between [3,5] and a depth size with the following values: [None,3,10,20].



(a)

(b)

Figure 4.2: CV error (a) and out-of-bag-error (b) vs number of features in the model. The optimum value is printed by the red dot

After testing different combinations, we have obtained the best result for a model consisting of 150 trees, 4 features (RM, DIS, PTRATIO and AGE) and 10 in the depth.

MSE-s in and out of sample for the last model

Error	in-sample	out-sample
MSE	3.593	22.739

Chapter 5

Conclusions

In this section we will present the conclusions drawn from each part of the project.

The work has been divided into two clear parts.

In the first part we have analysed the Boston dataset and made the necessary changes in the data to maximise our prediction results in the following section. Here, as we have seen, some of the data we have been working with have shown a certain number of outliers. Because of this, we have decided to suppress or re-evaluate these values in order to avoid problems in the following sections. Initially we started the analysis with 506 data for each variable. However, after this process the amount of data has decreased by 20%.

On the other hand, based on the p-value we have come to the conclusion that only 10 of the 12 variables are significant in explaining the variation of our response variable.

Using these 10 variables (characteristics) we have discarded the variables with a correlation between them greater than 70% and therefore we have finished the analysis of the dataset with 8 variables. 4 less than at the beginning.

In the second part we have build two different models in order to predict the variable MEDV: decision trees and random forest.

From the first we have concluded that the hyperparmeter tuning is crucial in order to avoid the overfiting. We have trained the model without modifications in the parameters and we have obtained a perfect model for the training data which means that the model was totally overfitting. However, making changes in the depth of the tree we have obtained the best results for a depth of 4 with a MSE (out of sample) of 45.005.

With the second model, we have followed the same strategy. However, the result for a dummy model, with all the parameters with their respective default values, has left as a much better result comparing with the one of the tuned decision tree. After that, we have combined different set of parameter, and we have conclude that the best result is suitable for 150 trees, 4 variables and depth equal to 10. The final MSE for the best model was 22.739. We have therefore concluded that the best fitting model for our prediction is the random forest.

Bibliography

[1] Kaggle.com. 2022. The Boston Housing Dataset. [online] Available at: https://www.kaggle.com/prasadperera/the-boston-housing-dataset [Accessed 2 January 2022].

[2] Müller, A. C., Guido, S. (2016). Introduction to Machine Learning with Python. Van Duuren Media.

[3] Kaggle.com. 2022. Boston Housing Data Analysis Machine Learning. [online] Available at: https://www.kaggle.com/mohitgoyal522/boston-housing-data-analysis-machine-learning [Accessed 2 January 2022].

[4] Medium. 2022. Feature selection — Correlation and P-value. [online] Available at: https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf [Accessed 2 January 2022].

[5] scikit-learn. 2022. OOB Errors for Random Forests. [online] Available at:<https://scikit-learn.org/stable/auto_examples/ensemble/plot_ensemble_oob.html> [Accessed5January2022].